Python Entrouvert - Development #5106

Gestion des tenants "sans modèle"

07 juillet 2014 11:34 - Thomas Noël

Statut:	Fermé	**Début:**	07 juillet 2014
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	100%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

django-tenant-schema doit faire l'association entre un domaine et un schéma pgsql.

Technique de base : utilisation d'un modèle Django.

Technique proposée dans ce ticket :

- quand un fichier /etc/logiciel/tenants/domaine.org/tenant existe, il indique qu'un tenant nomme "tenant" existe pour le domaine "domaine.org" pour le logiciel "logiciel"
- si un "settings.json" existe dans ce répertoire il permet de construire un SETTINGS pour le tenant
- ... et aussi un éventuel settings.py

Demandes liées:

Dupliqué par Hobo - Development #4981: Management command to create tenants f...	**Rejeté**	**17 juin 2014**

Révisions associées

Révision e4332b43 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Remove dead import on tenant_schemas.utils

refs #5106

Révision 73b85f5d - 12 septembre 2014 12:02 - Benjamin Dauvergne

In FileSystemLoader rename settings from MULTITENANT_TEMPLATE_DIRS to TENANT_TEMPLATE_DIRS to unformize with other settings

refs #5106

Révision 392340cf - 12 septembre 2014 12:02 - Benjamin Dauvergne

In FileSystemLoader use the schema name instead of the domain name for building template path

Also add a templates/ suffix.

refs #5106

Révision ec0613c1 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Add middleware to load settings from a JSON file based on the tenant

- Loaded settings are cached based on the mtime of the setting file
- JSON file path is <settings.TENANT_BASE>/<schema_name>/settings.json

refs #5106

Révision aae80ef0 - 12 septembre 2014 12:02 - Benjamin Dauvergne

New TenantMiddleware which try to find tenants based on the filesystem

If path <settings.TENANT_BASE>/<hostname>/schema exists, read this file an build
tenant modle with Tenant(domain_url=<hostname>, schema_name=file(<path>).read()).

refs #5106

Révision 4cea64a7 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Remove create-tenant command

refs #5106

Révision e8304bf1 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Make tenant model non savable

refs #5106

Révision ccec1ff4 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Import django-tenant-schemas commands to adapt them to our way of managing tenants

refs #5106

Révision 3afe385b - 12 septembre 2014 12:02 - Benjamin Dauvergne

Add command to create missing tenant schemas

refs #5106

Révision 5c1d1ad6 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Add command get_tenant_by_domain

refs #5106

Révision fbed23e8 - 12 septembre 2014 12:02 - Benjamin Dauvergne

Add command list_tenants (fixes #5044)

refs #5106

Révision b7f756de - 12 septembre 2014 12:02 - Benjamin Dauvergne

Add tests on multitenants features (fixes #5106)

Historique

#1 - 07 juillet 2014 11:48 - Benjamin Dauvergne

Plusieurs points de vigilance:

- refaire le middleware tenant_schemas/middleware.py qui dépend explicitement de get_tenant_model(),
- refaire les commandes migrate_schemas et sync_schemas qui parcourent la liste des schémas via un get_tenant_model().objects.all().

La commande tenant_command par contre ne dépend du modèle et peut-être utiliser telle quelle. Elle permet d'appeler n'importe quelle commande dans le contexte d'un tenant.

#2 - 07 juillet 2014 13:12 - Frédéric Péters

Côté portail admin, ça veut aussi dire refaire les formulaires de création/édition d'un tenant, pour ne plus dépendre de ModelForm, pareil pour les vues associées.

#3 - 07 juillet 2014 13:15 - Benjamin Dauvergne

Je ne crois pas qu'on soit obligé, coté portail-admin, de faire pareil.

#4 - 07 juillet 2014 13:35 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> La commande tenant_command par contre ne dépend du modèle et peut-être utiliser telle quelle. Elle permet d'appeler n'importe quelle commande dans le contexte d'un tenant.

En fait j'ai tort il y a aussi un problème: les commandes seront bien appelé dans le contexte de la base au niveau base de donnée mais concernant les settings comme on est pas passé dans le middleware des settings, ce n'est pas la bonne version qui en sera chargé donc il faudrait aussi prévoir au niveau des commandes le chargement des settings.

#5 - 07 juillet 2014 17:26 - Frédéric Péters

- Projet changé de Portail admin à Hobo

#6 - 30 juillet 2014 18:22 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#7 - 11 août 2014 13:30 - Frédéric Péters

Ça serait bien que ça arrive rapidement, c'est notamment un préalable à l'installation d'un passerelle en SaaS, ce qui serait utile pour Vincennes.

#8 - 19 août 2014 11:03 - Benjamin Dauvergne

- Statut changé de Nouveau à En cours

Finalement je complète/modifie ce qui est déjà dans python-entrouvert plutôt que de tout mettre dans hobo, hobo c'est finalement une application particulière.

Ce que je suis en train de faire:

- ajoute d'un settings TENANT_BASE
- si os.path.join(TENANT_BASE, <domain>, 'schema') existe on lit ce fichier et son contenu est le nom du tenant
- si os.path.join(TENANT_BASE, <domain>, 'settings.json' ou 'settings.py') existe ils sont chargés (l'ordre dépend de l'ordre de chargement des deux middleware concernés)

Par ailleurs le template loader a été modifié, pour charger tous les répertoires dans TENANT_TEMPLATE_DIRS, comme cela:

```
    for template_dir in settings.TENANT_TEMPLATE_DIRS:
        os.path.join(template_dir, <domain>, 'templates')
```

Ainsi on peut définir TENANT_TEMPLATE_DIRS = ('/var/lib/<logiciel>/tenants/', '/usr/share/<logiciel>/tenants/',) et pouvoir charger des templates venant d'un paquet et aussi d'une bidouille locale.

J'hésite à remplace <domain> un peu partout par le nom directement du schéma et déplacer le mapping nom de domain->schéma dans la configuration des virtual host en utilisant un entête HTTP maison nommé X-Tenant:. Dans ce cas au lieu d'un fichier /etc/logiciel/tenants/<domain>/schema qui contient le nom du schéma on a directement @/etc/logiciel/tenants/<schema>/ et le mapping des domain vers les schémas se fait dans nginx.

Qu'en pensez vous ?

#9 - 19 août 2014 11:15 - Frédéric Péters

Benjamin Dauvergne a écrit :

> Finalement je complète/modifie ce qui est déjà dans python-entrouvert plutôt que de tout mettre dans hobo, hobo c'est finalement une application particulière.

Ok; on déplace ce ticket ou tu crées un nouveau ?

> J'hésite à remplace <domain> un peu partout par le nom directement du schéma et déplacer le mapping nom de domain->schéma dans la configuration des virtual host en utilisant un entête HTTP maison nommé X-Tenant:. Dans ce cas au lieu d'un fichier /etc/logiciel/tenants/<domain>/schema qui contient le nom du schéma on a directement @/etc/logiciel/tenants/<schema>/ et le mapping des domain vers les schémas se fait dans nginx.
>
> Qu'en pensez vous ?

Le portail admin il a le nom de domaine comme info; de là l'agent peut lire os.path.join(TENANT_BASE, <domain>, 'schema'), c'est facile.

Dans l'autre option, à partir du nom de domaine, pour trouver le tenant associé, tu verrais ça comment ? (j'ai quelques idées mais rien de bien propre)

#10 - 19 août 2014 11:18 - Benjamin Dauvergne

Je suis en train de me dire que pour simplifier tout ça je pourrais avoir un simple TENANT_DIRS qui configure tout à la fois, les tenants, leur configuration et leur templates.

On aurait donc:

```
TENANT_DIRS = ('/etc/<logiciel>/tenants/',
'/var/lib/<logiciel>/tenants/',
'/usr/share/<logiciel>/tenants/')
```

#11 - 19 août 2014 11:22 - Benjamin Dauvergne

- Projet changé de Hobo à Python Entrouvert

Frédéric Péters a écrit :

> Benjamin Dauvergne a écrit :
>
> > Finalement je complète/modifie ce qui est déjà dans python-entrouvert plutôt que de tout mettre dans hobo, hobo c'est finalement une application particulière.
>
> Ok; on déplace ce ticket ou tu crées un nouveau ?

Je déplace.

> J'hésite à remplace <domain> un peu partout par le nom directement du schéma et déplacer le mapping nom de domain->schéma dans la configuration des virtual host en utilisant un entête HTTP maison nommé X-Tenant:. Dans ce cas au lieu d'un fichier /etc/logiciel/tenants/<domain>/schema qui contient le nom du schéma on a directement @/etc/logiciel/tenants/<schema>/ et le mapping des domain vers les schémas se fait dans nginx.
>
> Qu'en pensez vous ?

> Le portail admin il a le nom de domaine comme info; de là l'agent peut lire os.path.join(TENANT_BASE, <domain>, 'schema'), c'est facile.
>
> Dans l'autre option, à partir du nom de domaine, pour trouver le tenant associé, tu verrais ça comment ? (j'ai quelques idées mais rien de bien propre)

Effectivement coté portail admin on ne fonctionne pas du tout via nom de schéma. Dans ce cas je serai enclin à générer directement le nom de schéma en fonction du nom de domaine à la hostname.replace('.', '_').replace('-', '_') pour limiter encore plus la configuration et supprimer ce fichier schema. Le code ce serait alors:

```
    if not os.path.exists(os.path.join(TENANT_BASE, hostname)):
        raise TenantNotFound
    return Tenant(schema_name=hostname.replace('.', '_').replace('-', '_'), domain_url=hostname)
```

#12 - 19 août 2014 11:26 - Frédéric Péters

Oui, avoir une fonction déterminée pour faire la correspondance domaine→tenant, ça me va très bien.

#13 - 19 août 2014 11:44 - Jérôme Schneider

Ca me dérange vraiment qu'on se retrouve avec un nom de schéma généré depuis le hostname. Ca semble pratique à première vu mais je pense que ça va nous poser des problèmes pour les différents déploiements en prod, test et dev. On ne pourra pas copier les bases facilement et il faudra générer plein de symlinks dans les paquets pour les différents hostname (dev, test et prod).
On pourrait utiliser la technique de Benj avec un header HTTP ou alors on pourrait imaginer de mettre un mapping hostname -> nom du schéma / tenant dans un fichier de configuration générique ou encore dans chaque dossier de conf de chaque schéma on ajouterait une entrée hostnames.

#14 - 19 août 2014 11:48 - Benjamin Dauvergne

Jérôme Schneider a écrit :

> Ca me dérange vraiment qu'on se retrouve avec un nom de schéma généré depuis le hostname. Ca semble pratique à première vu mais je pense que ça va nous poser des problèmes pour les différents déploiements en prod, test et dev. On ne pourra pas copier les bases facilement et il faudra générer plein de symlinks dans les paquets pour les différents hostname (dev, test et prod).
> On pourrait utiliser la technique de Benj avec un header HTTP ou alors on pourrait imaginer de mettre un mapping hostname -> nom du schéma / tenant dans un fichier de configuration générique ou encore dans chaque dossier de conf de chaque schéma on ajouterait une entrée hostnames.

Ouaip c'est un vrai souci, les contournements que je vois:

- concernant les templates et leurs paquets, utiliser des liens symboliques avec les différents hostname de prod, preprod, etc...
- pour les migrations de données, penser à faire un DROP DATABASE <prod>; ALTER DATABASE <preppod> RENAME TO <prod> et inversement pour la reprise en preprod des données de prod.

#15 - 19 août 2014 11:57 - Frédéric Péters

Je notais :

> avoir une fonction déterminée pour faire la correspondance domaine→tenant, ça me va très bien

Ça peut très bien être mapping.get(hostname, hostname.replace('.', '_').replace('-', '_')), et que mapping vienne d'un json posé quelque part (ou tout autre format), non ?

#16 - 19 août 2014 15:18 - Jérôme Schneider

Frédéric Péters a écrit :

> Je notais :
>
> > avoir une fonction déterminée pour faire la correspondance domaine→tenant, ça me va très bien
>
> Ça peut très bien être mapping.get(hostname, hostname.replace('.', '_').replace('-', '_')), et que mapping vienne d'un json posé quelque part (ou tout autre format), non ?

C'est une très bonne solution. C'est ma préféré pour le moment.

Après une petite remarque pour Benjamin : je n'aime pas spécialement le fait d'avoir deux formats de fichier de configuration (json et python en l'occurrence) qui configure la même chose. Si on ne pouvait en garder qu'un ça serait parfait. Le portail citoyen v2 actuel qui prend yaml, json, python et les variables d'environnements c'est vraiment une cause de confusion. (je sais on en avait déjà parlé mais je préfère en remettre une couche).

#17 - 19 août 2014 15:25 - Benjamin Dauvergne

On en a déja parlé effectivement et on a conclu qu'on ne faisait plus que JSON et Python, ce que je fais là.

#18 - 19 août 2014 19:42 - Thomas Noël

Benjamin Dauvergne a écrit :

> Je suis en train de me dire que pour simplifier tout ça je pourrais avoir un simple TENANT_DIRS qui configure tout à la fois, les tenants, leur configuration et leur templates.
> On aurait donc: TENANT_DIRS = ('/etc/<logiciel>/tenants/', '/var/lib/<logiciel>/tenants/', '/usr/share/<logiciel>/tenants/')

Moi je mettrais le /var en dernier, mais sinon, ça me plait bien.

#19 - 19 août 2014 19:49 - Thomas Noël

Frédéric Péters a écrit :

> Je notais :
>
> > avoir une fonction déterminée pour faire la correspondance domaine→tenant, ça me va très bien
>
> Ça peut très bien être mapping.get(hostname, hostname.replace('.', '_').replace('-', '_')), et que mapping vienne d'un json posé quelque part (ou tout autre format), non ?

Oui, un explicite /etc/<logiciel>/tenant_by_domain.json (et si ce fichier n'existe pas, c'est pas grave).

Ensuite, pour toute les mécaniques devop, il faudra qu'on s'ajoute des petites commandes du style manage.py get_tenant_by_domain <domain> qui nous dira le nom du tenant sans qu'on ait besoin de le chercher "à la main".

#20 - 19 août 2014 19:52 - Thomas Noël

Benjamin Dauvergne a écrit :

- si os.path.join(TENANT_BASE, <domain>, 'settings.json' ou 'settings.py') existe ils sont chargés (l'ordre dépend de l'ordre de chargement des deux middleware concernés)
  (...)
  Qu'en pensez vous ?

Pour moi, faire en sorte que le settings.py soit toujours lancé *après* la prise en charge des settings.json

#21 - 20 août 2014 15:11 - Benjamin Dauvergne

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Je suis en train de me dire que pour simplifier tout ça je pourrais avoir un simple TENANT_DIRS qui configure tout à la fois, les tenants, leur configuration et leur templates.
> > On aurait donc: TENANT_DIRS = ('/etc/<logiciel>/tenants/', '/var/lib/<logiciel>/tenants/', '/usr/share/<logiciel>/tenants/')
>
> Moi je mettrais le /var en dernier, mais sinon, ça me plait bien.

Finalement je n'ai pas suivi cette piste parce que combiné des settings depuis plusieurs répertoires tout en gérant du cache c'était compliqué, j'y repenserai si le besoin s'en fait sentir.

#22 - 20 août 2014 15:14 - Benjamin Dauvergne

Thomas Noël a écrit :

> Oui, un explicite /etc/<logiciel>/tenant_by_domain.json (et si ce fichier n'existe pas, c'est pas grave).

Finalement c'est un dico dans les settings nommé TENANT_MAPPING.

#23 - 20 août 2014 15:15 - Benjamin Dauvergne

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> - si os.path.join(TENANT_BASE, <domain>, 'settings.json' ou 'settings.py') existe ils sont chargés (l'ordre dépend de l'ordre de chargement des deux middleware concernés)
> (...)
> Qu'en pensez vous ?
>
> Pour moi, faire en sorte que le settings.py soit toujours lancé *après* la prise en charge des settings.json

Ça dépend de l'ordre dans lequel on déclare le PythonSettingsMiddleware et le JSONSettingsMiddleware donc oui c'est faisable.

#24 - 20 août 2014 15:23 - Benjamin Dauvergne

- Fichier 0001-Remove-dead-import-on-tenant_schemas.utils.patch ajouté

- Fichier 0002-In-FileSystemLoader-rename-settings-from-MULTITENANT.patch ajouté

- Fichier 0003-In-FileSystemLoader-use-the-schema-name-instead-of-t.patch ajouté

- Fichier 0004-Add-middleware-to-load-settings-from-files-based-on-.patch ajouté

- Fichier 0005-New-TenantMiddleware-which-try-to-find-tenants-based.patch ajouté

- Fichier 0006-Remove-create-tenant-command.patch ajouté

- Fichier 0007-Make-tenant-model-non-savable.patch ajouté

- Fichier 0008-Import-django-tenant-schemas-commands-to-adapt-them-.patch ajouté

- Fichier 0009-Add-command-to-create-missing-tenant-schemas.patch ajouté

- Fichier 0010-Add-command-get_tenant_by_domain.patch ajouté

- Patch proposed changé de Non à Oui

Voilà tout le code, testé avec authentic et ce patch aux settings de celui-ci:

```
commit 302fda931113d7029864a19aeefce24654cb4563
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Wed Aug 20 14:37:14 2014 +0200

    Change settings for quick multitenant support

diff --git a/authentic2/settings.py b/authentic2/settings.py
index 5c9659c..8ed81ed 100644
--- a/authentic2/settings.py
+++ b/authentic2/settings.py
@@ -65,9 +65,9 @@ MANAGERS = ADMINS

 DATABASES = {
     'default': {
-        'ENGINE': 'django.db.backends.sqlite3',
-        'NAME': os.path.join(PROJECT_DIR, PROJECT_NAME + '.db'),
-    }
+        'ENGINE': 'tenant_schemas.postgresql_backend',
+        'NAME': 'authentic2',
+    },
 }

 for key in os.environ:
```

```
@@ -88,6 +88,7 @@ STATIC_URL = os.environ.get('STATIC_URL', '/static/')

 if DEBUG:
     TEMPLATE_LOADERS = (
+        'entrouvert.djommon.multitenant.template_loader.FilesystemLoader',
         'django.template.loaders.filesystem.Loader',
         'django.template.loaders.app_directories.Loader',
     )
@@ -110,6 +111,8 @@ TEMPLATE_CONTEXT_PROCESSORS = (
 )

 MIDDLEWARE_CLASSES = (
+    'entrouvert.djommon.multitenant.middleware.TenantMiddleware',
+    'entrouvert.djommon.multitenant.middleware.PythonSettingsMiddleware',
     'authentic2.middleware.LoggingCollectorMiddleware',
     'django.middleware.common.CommonMiddleware',
     'django.middleware.http.ConditionalGetMiddleware',
@@ -186,10 +189,31 @@ INSTALLED_APPS = (
     'authentic2.disco_service',
     'authentic2.manager',
     'authentic2',
+    'entrouvert.djommon.multitenant',
 )

 INSTALLED_APPS = plugins.register_plugins_installed_apps(INSTALLED_APPS)

+TENANT_APPS = INSTALLED_APPS
+
+TENANT_MODEL = 'multitenant.Tenant'
+TENANT_BASE = '/tmp/tenants/'
+TENANT_TEMPLATE_DIRS = (
+        '/tmp/tenants/',
+)
+
+SOUTH_DATABASE_ADAPTERS = {
+            'default': 'south.db.postgresql_psycopg2',
+}
+
+SHARED_APPS = (
+    'django.contrib.staticfiles',
+    'django.contrib.auth',
+    'django.contrib.contenttypes',
+    'django.contrib.sessions',
+    'django.contrib.messages',
+)
+
 MESSAGE_STORAGE = 'django.contrib.messages.storage.session.SessionStorage'
```

Pour initialiser:

```
$ mkdir /tmp/tenants
```

Pour créer un tenant:

```
$ echo 127.0.0.1 test.localhost >>/etc/hosts
$ mkdir /tmp/tenants/test_localhost
$ authentic2/run.sh create_schema
```

Pour définir des settings locaux:

```
$ echo A2_CAN_RESET_PASSWORD = False >/tmp/tenants/test_localhost/settings.py
```

Pour définir un template de page de login local:

```
$ mkdir -p /tmp/tenants/test_localhost/templates/auth/
$ cp authentic2/authentic2/templates/auth/login.html /tmp/tenants/test_localhost/templates/auth/
$ vi /tmp/tenants/test_localhost/templates/auth/login.html
```

Pour appliquer les migrations aux tenants

```
$ authentic2/run.sh migrate_schemas --tenant
```

Reste à faire les derniers ajustements ensuite on pourra l'utiliser dans nos différentes applications. J'ai retiré mes pull request sur django-tenant-schemas parce que je ne pense pas que ce soit acceptable en l'état pour le mainteneur et leur vitesse d'avancement n'est pas compatible avec notre calendrier, mais je referai certainement une tentative plus tard.

#25 - 20 août 2014 15:46 - Frédéric Péters

En vrac.

> [...] ce patch aux settings de celui-ci

On est d'accord pour dire que la config en tenants, elle vient par dessus nos applications (quand elles sont déployées, donc a priori dans le paquet .deb), mais qu'on ne modifie pas celles-ci pour l'utiliser dans leurs configs par défaut ?

```
 if DEBUG:
     TEMPLATE_LOADERS = (
+        'entrouvert.djommon.multitenant.template_loader.FilesystemLoader',
```

(sans avoir la totalité du fichier, ça fait bizarre d'avoir ça sous un "if DEBUG:")

~~

Une mise à jour de python-entrouvert sur un serveur avec des applications utilisant les tenants, ça va passer sans peine ? (on s'en fout si on n'a aucune installation utilisant des tenants pour le moment, c'est juste que je ne suis pas sûr de ça)

~~

0008-Import-django-tenant-schemas-commands-to-adapt-them-.patch, c'est sympa d'ajouter une petite note en haut des fichiers pour rappeler leur origine.

Et tant qu'à les adapter, je vois notamment ce bout qui pourrait aussi l'être :

```
        if not all_tenants:
            raise CommandError("""There are no tenants in the system.
To learn how create a tenant, see:
https://django-tenant-schemas.readthedocs.org/en/latest/use.html#creating-a-tenant""")
```

~~

```
$ mkdir /tmp/tenants/test_localhost
$ authentic2/run.sh create_schema
```

En traduisant ça vers du code générique pour le portail admin, ça donnerait :

```
$tenant = $app_manage get_tenant_by_domain $host
mkdir /var/lib/$app/$tenant
echo XXX > /var/lib/$app/$tenant/settings.json
$app_manage create_schema
```

Correct ?

~~

> Finalement c'est un dico dans les settings nommé TENANT_MAPPING.

Je pense que j'aurais préféré un fichier indépendant, que ça aurait été plus simple le jour où on voudra pouvoir ajouter un tenant sans redémarrer l'application.

#26 - 20 août 2014 16:22 - Benjamin Dauvergne

Frédéric Péters a écrit :

> En vrac.
>
> > [...] ce patch aux settings de celui-ci
>
> On est d'accord pour dire que la config en tenants, elle vient par dessus nos applications (quand elles sont déployées, donc a priori dans le paquet .deb), mais qu'on ne modifie pas celles-ci pour l'utiliser dans leurs configs par défaut ?
>
> [...]
>
> (sans avoir la totalité du fichier, ça fait bizarre d'avoir ça sous un "if DEBUG:")

J'ai modifié les settings d'authentic pour le test pas pour le commiter, et je l'ai recopié ici pour référence.

> Une mise à jour de python-entrouvert sur un serveur avec des applications utilisant les tenants, ça va passer sans peine ? (on s'en fout si on n'a aucune installation utilisant des tenants pour le moment, c'est juste que je ne suis pas sûr de ça)

Non c'est une mise à jour incompatible mais je peux restaurer ce que j'ai cassé volontairement:

- la classe Tenant que j'ai rendu inopérante pour ne pas se retrouver avec des modèles créés quand on n'en veut pas
- la classe EOTenantMiddleware qui chargeait des settings depuis la base
- restauré l'ancien FilesystemLoader qui n'ajoutait pas templates/ à la fin du chemin des templates pour un tenant et utilisait MULTITENANT_TEMPLATE_DIRS comme variable et pas TENANT_TEMPLATE_DIRS, mais dans ce cas il faudrait que je créé un autre classe avec un nom différent pour le nouvelle version

  ~~

  0008-Import-django-tenant-schemas-commands-to-adapt-them-.patch, c'est sympa d'ajouter une petite note en haut des fichiers pour rappeler leur origine.

Ok

> Et tant qu'à les adapter, je vois notamment ce bout qui pourrait aussi l'être :
>
> [...]

Je n'ai pas vraiment de doc sur laquelle pointer pour l'instant mais je vais mettre un commentaire du style "create a directory in settings.TENANT_BASE".

> ~~
>
> [...]
>
> En traduisant ça vers du code générique pour le portail admin, ça donnerait :
>
> [...]
>
> Correct ?

Non mais je vais corriger get_tenant_by_domain, pour l'instant il ne retourne la bonne valeur que si le répertoire du tenant existe déjà.

> ~~
>
> > Finalement c'est un dico dans les settings nommé TENANT_MAPPING.
>
> Je pense que j'aurais préféré un fichier indépendant, que ça aurait été plus simple le jour où on voudra pouvoir ajouter un tenant sans redémarrer l'application.

Les settings sont relus sur un kill -HUP du process gunicorn parent. Je ne suis pas fan de multiplier les fichiers de configuration, surtout qu'on peut simuler ça avec les outils actuels, par exemple en faisant:

```
TENANT_MAPPING = json.load(file('/etc/<logiciel>/tenants/tenant_mapping.conf'))
```

#27 - 20 août 2014 16:45 - Benjamin Dauvergne

L'autre souci c'est que pour l'instant l'exécution d'une commande donne ça:

```
(authentic2-venv)bdauvergne@fenouil:~/Code/authentic2$ ./run.sh get_tenant_by_domain test.localhost
Cannot access global cache path, using in project cache directory /home/bdauvergne/Code/authentic2/authentic2/
../cache

Warning: You should put 'tenant_schemas' at the end of INSTALLED_APPS like this:
INSTALLED_APPS = TENANT_APPS + SHARED_APPS + ('tenant_schemas',)
This is neccesary to overwrite built-in django management commands with their schema-aware implementations.

test_localhost
```

Le premier message ça ira mais le deuxième n'est pas aussi simple à faire taire.

#28 - 20 août 2014 16:53 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> L'autre souci c'est que pour l'instant l'exécution d'une commande donne ça:
>
> [...]

> Le premier message ça ira mais le deuxième n'est pas aussi simple à faire taire.

J'ai fait une pull request pour ça upstream.

https://github.com/bernardopires/django-tenant-schemas/pull/171

#29 - 20 août 2014 17:06 - Thomas Noël

Benjamin Dauvergne a écrit :

> Thomas Noël a écrit :
>
> > Oui, un explicite /etc/<logiciel>/tenant_by_domain.json (et si ce fichier n'existe pas, c'est pas grave).
>
> Finalement c'est un dico dans les settings nommé TENANT_MAPPING.

Je trouve ça dommage, ça va demander un "restart" de l'application alors que le reste était dynamique.

#30 - 20 août 2014 17:13 - Thomas Noël

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Thomas Noël a écrit :
> >
> > > Oui, un explicite /etc/<logiciel>/tenant_by_domain.json (et si ce fichier n'existe pas, c'est pas grave).
> >
> > Finalement c'est un dico dans les settings nommé TENANT_MAPPING.
>
> Je trouve ça dommage, ça va demander un "restart" de l'application alors que le reste était dynamique.

... et je viens de lire que ça a déjà été discuté, ok pour le "kill -HUP", considérant qu'en mode normal de déploiement, on ne devrait pas avoir besoin d'un TENANT_MAPPING.

#31 - 20 août 2014 17:20 - Frédéric Péters

Thomas Noël a écrit :

> ... et je viens de lire que ça a déjà été discuté, ok pour le "kill -HUP", considérant qu'en mode normal de déploiement, on ne devrait pas avoir besoin d'un TENANT_MAPPING.

Et je confirme ici que le kill -HUP ça me va très bien aussi.

#32 - 20 août 2014 17:42 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Frédéric Péters a écrit :
>
> > Une mise à jour de python-entrouvert sur un serveur avec des applications utilisant les tenants, ça va passer sans peine ? (on s'en fout si on n'a aucune installation utilisant des tenants pour le moment, c'est juste que je ne suis pas sûr de ça)

Et sur ça je laisse béton ?

> Non c'est une mise à jour incompatible mais je peux restaurer ce que j'ai cassé volontairement:
>
> - la classe Tenant que j'ai rendu inopérante pour ne pas se retrouver avec des modèles créés quand on n'en veut pas
> - la classe EOTenantMiddleware qui chargeait des settings depuis la base
> - restauré l'ancien FilesystemLoader qui n'ajoutait pas templates/ à la fin du chemin des templates pour un tenant et utilisait MULTITENANT_TEMPLATE_DIRS comme variable et pas TENANT_TEMPLATE_DIRS, mais dans ce cas il faudrait que je créé un autre classe avec un nom différent pour le nouvelle version

#33 - 20 août 2014 17:55 - Frédéric Péters

> Et sur ça je laisse béton ?

J'avais juste fait une réponse rapide sur le point, pour dire que j'étais d'accord avec Thomas aussi.

Sur les autres points; j'écrivais :

> (on s'en fout si on n'a aucune installation utilisant des tenants pour le moment, c'est juste que je ne suis pas sûr de ça)

Je maintiens ça; ça m'irait vraiment très bien de ne pas transporter de legacy. Tu sais s'il y a du multitenant quelque part dans des applications déployées en prod ? (le seul utilisateur "public" de tenant qui me vienne, c'est le poc univnautes in the cloud de Thomas, sur lequel on ne doit pas bloquer).

> Non mais je vais corriger get_tenant_by_domain, pour l'instant il ne retourne la bonne valeur que si le répertoire du tenant existe déjà.

Une fois cela ok, les quatre lignes que je donne pour le déploiement depuis un agent hobo sont donc correctes ? Si c'est le cas ça me va bien.

#34 - 21 août 2014 10:20 - Benjamin Dauvergne

- Fichier 0010-Add-command-get_tenant_by_domain.patch ajouté

Frédéric Péters a écrit :

> Sur les autres points; j'écrivais :
>
> > (on s'en fout si on n'a aucune installation utilisant des tenants pour le moment, c'est juste que je ne suis pas sûr de ça)
>
> Je maintiens ça; ça m'irait vraiment très bien de ne pas transporter de legacy. Tu sais s'il y a du multitenant quelque part dans des applications déployées en prod ? (le seul utilisateur "public" de tenant qui me vienne, c'est le poc univnautes in the cloud de Thomas, sur lequel on ne doit pas bloquer).

Non rien à ma connaissance et je n'ai pas connaissance non plus d'un poc univnautes in the cloud multitnenant pour l'instant.

> > Non mais je vais corriger get_tenant_by_domain, pour l'instant il ne retourne la bonne valeur que si le répertoire du tenant existe déjà.
>
> Une fois cela ok, les quatre lignes que je donne pour le déploiement depuis un agent hobo sont donc correctes ? Si c'est le cas ça me va bien.

Non il y a toujours le souci des lignes en sortie gênantes. Actuellement il faut faire comme ça (ajout de | tail -n1):

```
$tenant = $app_manage get_tenant_by_domain $host | tail -n1
mkdir /var/lib/$app/$tenant
echo XXX > /var/lib/$app/$tenant/settings.json
$app_manage create_schema
```

#35 - 21 août 2014 10:32 - Frédéric Péters

> Non il y a toujours le souci des lignes en sortie gênantes. Actuellement il faut faire comme ça (ajout de | tail -n1):

Ah oui, ça dans leur *init*.py, c'est moche :

```
if settings.INSTALLED_APPS[-1] != 'tenant_schemas':
    print recommended_config
```

Tu tentes une pull request ajoutant l'envoi vers sys.stderr ?

#36 - 21 août 2014 14:13 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Non il y a toujours le souci des lignes en sortie gênantes. Actuellement il faut faire comme ça (ajout de | tail -n1):

> Ah oui, ça dans leur *init*.py, c'est moche :

> [...]
>
> Tu tentes une pull request ajoutant l'envoi vers sys.stderr ?

Déjà fait #5106#note-28 (solution un peu différente).

#37 - 21 août 2014 17:12 - Thomas Noël

Benjamin Dauvergne a écrit :

> Non rien à ma connaissance et je n'ai pas connaissance non plus d'un poc univnautes in the cloud multitnenant pour l'instant.

Je confirme que ça a existé, mais on peut vraiment faire comme si ça n'existait pas (c'était un bidule sur https://univnautes-idp.dev.entrouvert.org/). Il n'y a pas de code historique à maintenir dans python-entrouvert au sujet des tenants (youpi).

#38 - 25 août 2014 10:22 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Frédéric Péters a écrit :
>
> > Non il y a toujours le souci des lignes en sortie gênantes. Actuellement il faut faire comme ça (ajout de | tail -n1):
>
> Ah oui, ça dans leur *init*.py, c'est moche :
>
> [...]
>
> Tu tentes une pull request ajoutant l'envoi vers sys.stderr ?
>
> Déjà fait #5106#note-28 (solution un peu différente).

Mon patch n'est pas évident j'ai du expliquer un peu https://github.com/bernardopires/django-tenant-schemas/pull/171#issuecomment-53238189

#39 - 28 août 2014 19:21 - Thomas Noël

Benjamin Dauvergne a écrit :

> Benjamin Dauvergne a écrit :
>
> > Frédéric Péters a écrit :
> >
> > > Ah oui, ça dans leur *init*.py, c'est moche :
> > > [...]
> > > Tu tentes une pull request ajoutant l'envoi vers sys.stderr ?
> >
> > Déjà fait #5106#note-28 (solution un peu différente).
>
> Mon patch n'est pas évident j'ai du expliquer un peu https://github.com/bernardopires/django-tenant-schemas/pull/171#issuecomment-53238189

Si ça passe pas, faudra re-tenter avec juste le >>sys.stderr et insister un peu..

Mais plus généralement, vu qu'on ne maîtrisera jamais la sortie du manage.py, on peut la formater de notre côté. Par exemple :

```
class Command(BaseCommand):
    help = "Create schemas"   # tiens d'ailleurs ça va pas, ça ;)

    def handle(self, *args, **options):
        for arg in args:
            print "get_tenant_by_domain %s: %s" % (arg, TenantMiddleware.hostname2schema(arg))
```

ce qui permet de faire un peu moins "magique" : manage.py get_tenant_by_domain mon.site | grep "get_tenant_by_domain mon.site:" | cut -f2 -d:

(enfin c'est surtout moins magique pour les vieux shellmen moi, on va dire).

#40 - 01 septembre 2014 09:56 - Frédéric Péters

> ce qui permet de faire un peu moins "magique" : manage.py get_tenant_by_domain mon.site | grep "get_tenant_by_domain mon.site:" | cut -f2 -d:

Je n'aime vraiment pas :/

#41 - 02 septembre 2014 15:34 - Benjamin Dauvergne

Frédéric Péters a écrit :

> > ce qui permet de faire un peu moins "magique" : manage.py get_tenant_by_domain mon.site | grep "get_tenant_by_domain mon.site:" | cut -f2 -d:
>
> Je n'aime vraiment pas :/

Pour moi le mieux pour l'instant vu qu'on package django-tenant-schemas à la main c'est de corriger à ce moment.

#42 - 02 septembre 2014 15:34 - Benjamin Dauvergne

- Fichier 0004-Add-middleware-to-load-settings-from-files-based-on-.patch supprimé

#43 - 02 septembre 2014 15:35 - Benjamin Dauvergne

- Fichier 0004-Add-middleware-to-load-settings-from-files-based-on-.patch ajouté

Nouvelle version du middleware pour charger des settings JSON.

#44 - 02 septembre 2014 15:35 - Benjamin Dauvergne

- Fichier 0008-Import-django-tenant-schemas-commands-to-adapt-them-.patch supprimé

#45 - 02 septembre 2014 15:36 - Benjamin Dauvergne

- Fichier 0008-Import-django-tenant-schemas-commands-to-adapt-them-.patch ajouté

Correction dans les commandes, notamment pour signaler la possibilité d'utiliser le domaine au lieu du nom de schéma.

#46 - 11 septembre 2014 12:57 - Benjamin Dauvergne

- Fichier 0004-Add-middleware-to-load-settings-from-a-JSON-file-bas.patch ajouté

J'ai viré le chargement depuis un fichier python et je suis revenu à l'ancienne version du chargement JSON, s'il faut combiner plusieurs fichiers mieux vaut que ce soit l'outil de configuration qui le fasse plutôt que d'avoir un algorithme compliqué au chargement.

#47 - 11 septembre 2014 14:12 - Frédéric Péters

C'était quand même utile d'avoir la possibilité de placer du Python par derrière, pour avoir la main sur des trucs qu'on n'arriverait pas à exprimer en json, ou pour poser une spécificité qu'on ne maitriserait pas via le déploiement.

#48 - 11 septembre 2014 16:00 - Benjamin Dauvergne

- Fichier 0004-Add-middleware-to-load-settings-from-a-JSON-file-bas.patch ajouté

New new patch, je vais pousser ensuite je pense.

#49 - 12 septembre 2014 12:01 - Benjamin Dauvergne

- Fichier 0012-Add-tests-on-multitenants-features.patch ajouté

Dernier commit avec des tests, ils passent tous sur passerelle.

#50 - 12 septembre 2014 12:05 - Benjamin Dauvergne

- Statut changé de En cours à Résolu (à déployer)

- % réalisé changé de 0 à 100

Appliqué par commit b7f756dee00990279833fb0427aeb61abcb452bb.

#51 - 16 septembre 2014 12:54 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-Remove-dead-import-on-tenant_schemas.utils.patch	832 octets	20 août 2014	Benjamin Dauvergne
0002-In-FileSystemLoader-rename-settings-from-MULTITENANT.patch	2,69 ko	20 août 2014	Benjamin Dauvergne
0003-In-FileSystemLoader-use-the-schema-name-instead-of-t.patch	1,87 ko	20 août 2014	Benjamin Dauvergne
0005-New-TenantMiddleware-which-try-to-find-tenants-based.patch	4,39 ko	20 août 2014	Benjamin Dauvergne
0006-Remove-create-tenant-command.patch	3,81 ko	20 août 2014	Benjamin Dauvergne
0007-Make-tenant-model-non-savable.patch	2,26 ko	20 août 2014	Benjamin Dauvergne
0009-Add-command-to-create-missing-tenant-schemas.patch	1,47 ko	20 août 2014	Benjamin Dauvergne
0010-Add-command-get_tenant_by_domain.patch	1,06 ko	20 août 2014	Benjamin Dauvergne
0010-Add-command-get_tenant_by_domain.patch	1,05 ko	21 août 2014	Benjamin Dauvergne
0004-Add-middleware-to-load-settings-from-files-based-on-.patch	5,5 ko	02 septembre 2014	Benjamin Dauvergne
0008-Import-django-tenant-schemas-commands-to-adapt-them-.patch	21,2 ko	02 septembre 2014	Benjamin Dauvergne
0004-Add-middleware-to-load-settings-from-a-JSON-file-bas.patch	3,89 ko	11 septembre 2014	Benjamin Dauvergne
0004-Add-middleware-to-load-settings-from-a-JSON-file-bas.patch	5,14 ko	11 septembre 2014	Benjamin Dauvergne
0012-Add-tests-on-multitenants-features.patch	4,39 ko	12 septembre 2014	Benjamin Dauvergne